

Hypothesis Generation Strategies for Adaptive Problem Solving*

Barbara Engelhardt, Steve Chien, Darren Mutz
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
{firstname.lastname}@jpl.nasa.gov

Abstract – Proposed missions to explore comets and moons will encounter environments that are hostile and unpredictable. Any successful explorer must be able to adapt to a wide range of possible operating conditions in order to survive. The traditional approach of constructing special-purpose control methods would require a information about the environment, which is not available a priori for these missions. An alternate approach is to utilize a general control approach with significant capability to adapt its behavior, a so called *adaptive problem-solving* methodology. Using adaptive problem-solving, a spacecraft can use reinforcement learning to adapt an environment-specific search strategy given the craft's general problem solver with a flexible control architecture. The resulting methods would enable the spacecraft increase its performance with respect to probability of survival and mission goals. We discuss an application of this approach to learn control strategies in planning and scheduling for three space mission models: Space Technologies 4, a Mars Rover, and Earth Observer One.

TABLE OF CONTENTS

1. INTRODUCTION
2. MOTIVATION FOR ADAPTIVE PROBLEM SOLVING
3. MOTIVATIONAL EXAMPLE
4. PLANNING SYSTEM
5. GENERATING CONTROL STRATEGIES
6. ADAPTIVE PROBLEM SOLVING
7. GENERATING HYPOTHESES
8. METHOD IMPLEMENTATION
9. EMPIRICAL EVALUATION
10. RELATED WORK
11. FUTURE WORK
12. CONCLUSIONS

INTRODUCTION

Proposed missions to explore comets and moons will encounter environments that are hostile and unpredictable. Because of light-time communication delays, these missions require an autonomous explorer that can adapt to handle possible environments. For autonomous planning systems, the high-level actions of the spacecraft must be planned with sufficient environmental information to ensure that the resulting plans are admissible. Generic control methods will not account for domain-specific features when operating a spacecraft. The spacecraft could easily be lost based on inappropriate behavior for the particular environment due to overly-generic control methods [8].

On the other hand, developing and testing domain-specific control methods is extremely difficult, and requires support of a domain expert. Moreover, the domain expert must have knowledge about the environment in which the spacecraft is operating, which is not available before the spacecraft arrives at the location to explore. If experts are not available, the spacecraft must be able to automatically adapt a flexible control structure specific to the new environment.

Adaptive problem solving addresses these problems by enabling the development and maintenance of effective control strategies without extensive domain-specific knowledge. An adaptive problem solver is given: (1) a generic set of control strategies and (2) a flexible control architecture, and uses a statistical method to estimate the quality of each control strategy or generate a more appropriate strategy. Adaptive problem solving also provides hard statistical guarantees on the quality of the behavior for each adapted control method. Using adaptive problem solving techniques, spacecraft exploration in unknown environments becomes feasible.

In this paper, we describe how adaptive problem solving can be used to adapt the control methods of a spacecraft in-situ without relying on domain expertise. The value of this method is empirically shown in the context of three spacecraft operations scheduling problems in a generic planning and scheduling environment. By adapting control strategies for each domain, the lifespan of the spacecraft is improved since the adaptive problem solver can increase chances of spacecraft survival and continue to update the control methods based on aging hardware or environmental changes.

Motivational Example

The comet lander will land on a surface of unknown density, with the goals of drilling into the comet 90% and imaging

its surroundings 10% of the time allocated to accomplishing the goals. Many situations will force these percentages to be adapted. One scenario might be that the surface of the comet is much denser than expected, so the rate of drilling is decreased and the wear on the drill is increased. The lander might decide to adjust its priorities to taking more images instead of drilling. Another scenario might be that drilling caused a layer of dust on the surface to drift up, the dust might limit the visibility of the lander. Taking images might be ineffective, so the lander would optimally delay its drilling activities until the dust settled, or put off taking images altogether. Failure to adapt to these situations could cost the lander the mission, by depleting resources too rapidly, not accomplishing mission objectives, or wearing out equipment. Not all possible situations can be enumerated before the mission; instead an adaptive problem solver checks the current control strategy's performance in the given environment and responds to changes by adapting the control strategy, independent of the cause of the change. An adaptive problem solver would continually adapt the control strategy if it found the current strategy non-optimal.

MOTIVATION FOR ADAPTIVE PROBLEM SOLVING

Selecting an effective control strategy in a specific domain from a set is difficult without information about how the strategies perform over a distribution of tasks in that domain. Although there exist classes of heuristics that are intended to be suitable for all planning domains, some amount of efficiency is sacrificed in generalizing the heuristics by failing to take advantage of the specific domain structure [8]. On the other hand, determining how each heuristic or set of heuristics performs in a problem domain can be costly. Adaptive problem solving attempts to estimate the performance of each strategy in a given domain by collecting samples of the strategy's performance over the problem distribution. The control strategies are represented as sets of heuristics so that they may be robust enough to perform well over the entire problem distribution even when they are slightly suboptimal, as opposed to a single heuristic which may not be as flexible. Some amount of generality is beneficial when domains are not known *a priori*, the domain structure changes rapidly, or a domain expert is not available and a complete strategy domain search is not possible.

PLANNING SYSTEM

The planning and scheduling system with a flexible control architecture used to evaluate the control strategies for each model is a version of the ASPEN (Automated Scheduling and Planning ENvironment) system [4]. ASPEN is a configurable, generic planning/scheduling application framework that can be tailored to specific domains to create conflict-free plans or schedules.

ASPEN employs planning and scheduling techniques to automatically generate a necessary activity sequence to achieve the input goals. This sequence is produced by utilizing an iterative repair algorithm [18] which classifies conflicts and attacks them each individually. Conflicts occur when a plan constraint has been violated where this constraint could be temporal or involve a resource, state or activity parameter. Conflicts are resolved by performing one or more schedule modifications such as moving, adding, or deleting activities

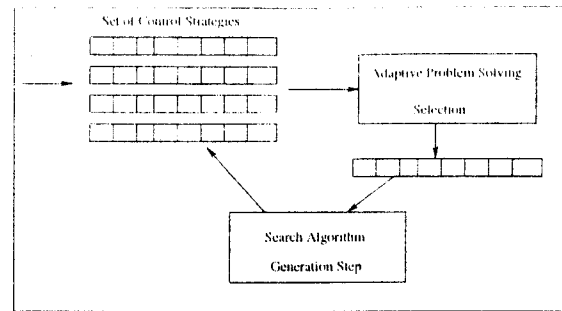


Figure 1: Hypothesis Generation Diagram

at a point in the search where a choice can be made by the scheduler, called a choice point. The target of the repair modification is chosen by a heuristic method. For each type of choice point, there exists a different set of heuristic methods to use in repair which can be modified easily.

The quality of a resulting schedule generated by ASPEN is measured by a set of preferences specified by the user. This set of preferences specifies the quality functions associated with certain metrics in the schedule, such as battery power usage or number of science goals achieved, and the possible cutoffs of the metric values. Although currently the ASPEN system does not take preferences into account while it performs iterative repair, this is a possible addition to the heuristics in future work.

GENERATING CONTROL STRATEGIES

Control strategies can be generated using search techniques and evaluated using adaptive problem solving. Given a set of control strategies, the adaptive problem solver selects the top strategy or strategies based on estimations of their quality parameters, and returns them to the search algorithm. The search algorithm produces the subsequent set of hypotheses using algorithm-specific techniques. The new set of strategies is passed to the adaptive problem solver for evaluation. This cycle continues until a certain amount of time has passed or another stopping criterion of the specific search algorithm has been met (see figure 1).

ADAPTIVE PROBLEM SOLVING

The adaptive problem solver attempts to select the top strategies from a set of strategies, supplied by the search algorithm, whose quality is a function of unknown environmental parameters. It makes estimates of the parameters for utility of a strategy and cost of a sample in order to achieve a requested accuracy for a statistical decision requirement, which is a function of the accuracy of each pair-wise comparison of set members. The adaptive problem solver iteratively refines the utility and cost parameter estimates by acquiring training examples at the estimated cost for each strategy (see figure 2). The normal parametric model for reasoning about statistical error is used in this analysis, which assumes that the difference between the expected utility and estimated utility of a hypothesis can be accurately approximated by a normal distribution. This assumption is grounded in the Central Limit Theorem and is further discussed in [2]. The analysis would

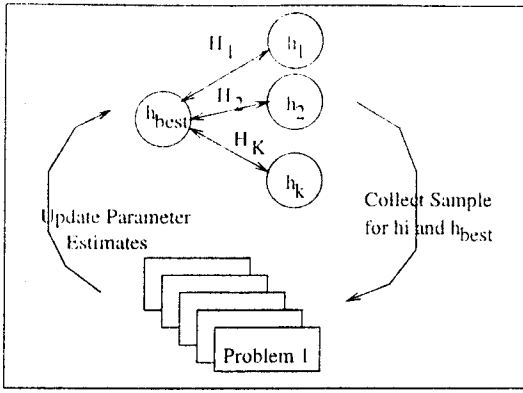


Figure 2: Adaptive Problem Solving Diagram

change given a different parametric model, but the results should be analogous for conventional models.

Since parameter estimates are refined by random sampling, it is impossible to place perfect accuracy requirements on the selection algorithms. In practice, probabilistic requirements, or *decision criteria*, on the relative accuracy of the parameter estimates (and subsequent strategy selection) are chosen as parameterized forms that allow a tradeoff between accuracy and cost.

Specifically, decision requirements take a set of hypotheses and a probabilistic error bound, and terminate when one of the hypotheses can be shown to have the greatest mean, evaluated through pair-wise comparisons, with a confidence specified by the given error bound. The overall error for selection is a function of the error of each pair-wise comparison. Rational analysis can be used to allocate error to each pairwise comparison in such a way as to attempt to optimize the resource usage necessary to acquire a sufficient number of samples to achieve the decision requirement.

In this analysis, the decision requirement that is used in the adaptive problem solver is the probably approximately correct (PAC) requirement. The approach of using adaptive solving with rational analysis to evaluate strategies has a natural correspondence in other decision requirements, and the choice of using PAC in this analysis is mostly based on their prevalence rather than specific attributes of the requirements themselves. An alternative decision requirement, the expected loss requirement, was evaluated compared with the PAC requirement and found to have minimal impact on the outcome.

PAC Requirement

In order to satisfy the PAC requirement, the hypothesis estimated to be the best must be within some user-specified constant ϵ distance of the true best hypothesis with probability $1 - \delta$. The sum of the error from each pair-wise comparison is bounded by this probability. Let H_{sel} be the expected utility of the selected hypotheses and H_i be the expected utility for the remaining hypotheses. Let \hat{H} be the estimate of the expected utility of a hypothesis. It is sufficient to bound the probability of error in selection for pair-wise comparisons with the following equation:

$$\sum_{i=1}^{k-1} Pr[\hat{H}_i < \hat{H}_{sel} - \epsilon | H_i > H_{sel} + \epsilon] \leq \delta \quad (1)$$

Thus the problem of bounding the overall error reduces to bounding the error of each $k - 1$ comparisons of the chosen best hypothesis to the rest of the hypotheses.

The normality assumption reduces equation 1 to a function of the parameter estimates, the number of examples n used to refine the estimates, the closeness parameter ϵ , and an unknown variance term σ^2 . The two stopping criteria for selection are *dominance*, which is based on achieving a probability (δ) through sampling that h_i will perform better on a specific problem than h_j , and *indifference*, which is the probability that the difference between performances will fall within ϵ of 0. For the rest of this discussion, ϵ is ignored to simplify understanding. The equation for the probability of incorrect selection for a pair-wise comparison, α_i , is:

$$\alpha_i = \Phi \left(- (H_{sel} - H_i) \frac{\sqrt{n}}{\sqrt{\sigma_{sel,i}^2}} \right) \quad (2)$$

We can use this relationship to determine how many training examples to allocate to each comparison, given the error bound on the probability of a mistake, an estimate of the difference in expected utility, and an estimate of the variance of each hypothesis:

$$n_{sel,i} = \frac{\sigma_{sel,i}^2}{(H_{sel} - H_i)^2} [\Phi^{-1}(\alpha_i)]^2 \quad (3)$$

Rational Analysis

The hypothesis selection algorithm as presented does not take advantage of unequal distribution of error. By distributing error unequally across the pair-wise comparisons using the estimates of the cost and utility parameters, we can attempt to satisfy the requirements using the minimum possible cost. The general idea of rational analysis is to choose the error α_i for each comparison to minimize, subject to the given decision requirements:

$$\sum_{i=1}^{k-1} C_{sel,i} n_{sel,i}$$

The algorithm must only ensure that the *sum* of the errors remains less than the given bound. If one pair-wise comparison requires many more samples to achieve the same amount of accuracy as another pair-wise comparison, then if the first comparison is allowed to have more error and the second is allowed less, the overall cost of achieving those local requirements might be reduced. In practice, this method significantly reduces the number of samples necessary to achieve the requirement for certain domains, as shown in [1].

GENERATING HYPOTHESES

In order to search the space of hypotheses, search algorithms are used to generate hypotheses and search the hypothesis domain for the highest scoring, or the set of highest scoring, hypotheses. At each level of search, an adaptive problem solving algorithm is used to evaluate the competing hypotheses with a

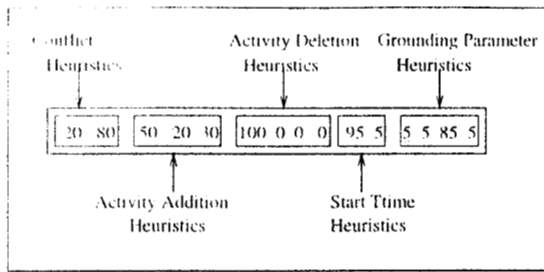


Figure 3: Hypothesis Vector Diagram

given confidence bound. We assume that time is a constrained resource as in [7, 3], so the search algorithms should limit the number of search levels, while at the same time allowing non-optimal choices so that the search will cover areas that could not be reached through simple hill-climbing.

Local Beam Search

The first algorithm that is used to generate and search over hypotheses is local beam search [11]. In a flexible planning and scheduling domain, each hypothesis, or combination of heuristics, can be represented as a vector of percentages where the percentages of heuristics associated with a certain type of choice point in ASPEN sum to 100% (see figure 3). A random heuristic is included for each plan problem. The basic algorithm is included below.

We chose a neighborhood of a vector to be defined as, for each subset of heuristics associated with a certain choice point, changing one of the usage percentages by a certain range, and scaling all of the other usage percentages equal amounts so that the sum is still 100% (see figure 3). Let l be the bound on the number of hypotheses the adaptive problem solver can evaluate.

Hill Climbing (initial set of hypotheses)

While (timeRemains)

Select top x hypotheses using PAC with confidence c .

Create $\frac{l}{b}$ higher scoring successors of top b hypotheses, where successors are generated in the neighborhood of original hypothesis.

End

Genetic Algorithm

The second algorithm that is used to learn hypotheses is a genetic algorithm [5]. Each hypothesis is represented as a vector of percentages, as in the local beam search. The three general operators (crossover, mutation, and reproduction) are used to generate the next set of hypotheses to search over, and ranking the hypotheses is done using adaptive problem solving. The crossover operator is not aware of the different subsets of heuristics, and may choose to split within one of those subsets. Mutation also works without knowledge of the constraint that subsets must sum to 100%, so each subset is

scaled to 100 uniformly after the mutation operator is run. The basic algorithm is shown below.

Genetic Algorithm (initial set of hypotheses)

While (timeRemains)

Rank x hypotheses using adaptive problem solving with Confidence c

Store out hypothesis with highest score.

Select parents from within highest ranked t with probability $ptop$

'Reproduce' using crossover with probability $pcross$ or reproduction with probability $(1 - pcross)$

Mutate offspring with probability $pmutation$

End

Search Considerations

In the context of this paper, both algorithms start the search with a set of human expert derived strategies that are currently in use in the domain model.

Both of these search techniques can be used for strict hill-climbing in the search space. But strict hill-climbing will limit the amount of space that is searched by restricting itself to a local maximum. Locally non-optimal steps are added to the search to possibly expand the breadth of the search.

There are many ways to allow locally non-optimal choices to be made during the search. One way is to allow the search algorithm to choose a hypothesis to propagate that scores worse than the hypotheses in the beam. Another way is to set the confidence bound in the adaptive problem solver's decision requirement low enough that it might include an incorrect hypothesis in its selection. This second method has the added benefit of decreasing the time associated with evaluating the hypotheses in practice. Both of these methods have the effect of including non-optimal steps into the search.

If a strict upper bound exists for time, and non-optimal choices are allowed, both of these algorithms can be implemented as anytime algorithms. The top ranking algorithms can be saved at each iteration of the search.

Although adaptive problem solving approximates the score, the ranking of each hypothesis on one search level is based on acquiring enough information about the performance of an hypothesis so that the ranking can be estimated within a required degree of accuracy for the pair-wise comparisons. This does not ensure that the set of hypotheses with the highest scores are ranked correctly, because their score is an approximation based on an error requirement that could be high. The set of top scoring hypotheses must be evaluated with a high confidence level before the absolute best can be selected.

Attributes of these two different search algorithms allow them to perform in different ways to provide insight into characteristics of the search space. Whereas the propagation of a vector in local beam search ignores potential dependencies between different subsets of heuristics, the crossover operation in genetic algorithms tends to propagate subsets of the string based on the distance between individual values in the vector,

known as the subset's *defining length* [5]. Based on where the subsets for heuristics of different types are located in the vector, the genetic algorithm may or may not reproduce them as a set. For this reason, genetic algorithms may be superior to local beam searches for domains where the influences of different heuristic types are dependent.

METHOD IMPLEMENTATION

An adaptive control system of this type can be used in mission operations in multiple capacities. It can be used from the start to design the spacecraft constraints and payload, by evaluating each of the potential designs against possible environments and comparing results. The system can be used on the ground to perform mission planning and during flight to quickly develop new schedules based on changing domains or spacecraft deterioration. The system might be used on-board a spacecraft to perform real-time fault detection and recovery. Environmental constraints for the spacecraft, such as the density or temperature of the surface for a lander, can be determined when they are available to the spacecraft. Accurate constraints are required for operation of a spacecraft in an unknown environment regardless of whether an automated planner is on-board the craft. These constraints can be used to update the on-board or ground-based model of the environment, and adaptive problem solving can be used to efficiently determine the optimal planning heuristics for the current environment.

EMPIRICAL EVALUATION

We claim that hypothesis generation can efficiently find a better set of hypotheses to produce high quality solutions in a given domain than an existing set, using adaptive problem solving to evaluate the performance of each hypothesis. In this section we provide evidence that in practice, these methods can generate heuristic sets superior to those generated by model experts. Furthermore, the generation methods are compared to evaluate how they perform for each given search domain.

The test of real-world applicability is based on three domains related to planned space missions, using the ASPEN planning and scheduling system. The original set of hypotheses that is used is the set of heuristic combinations currently in use in these and related models. We hope this illustrates how this type of method can be useful in real-world domains, by improving on control strategies already in use, improving the strategies during missions, or updating the strategies to handle domain shifts.

Evaluation

New Millennium EO-1 Domain – New Millennium Earth Observer 1 (EO-1) is an earth imaging satellite featuring an advanced multi-spectral imaging device. EO-1 mission operations consists of managing spacecraft operability constraints (power, thermal, pointing, buffers, consumables, telecommunications, etc.) and science goals (imaging of specific targets within particular observation parameters). Of particular difficulty is managing the downlinks as the amount of data generated by the imaging device is quite large and uplink opportunities are a limited resource. In addition, because science targets for EO-1 are based upon short-term cloud predictions, schedules must be generated daily. Automated planning

would supply needed assistance with daily scheduling, which is not feasible with EO-1's three person mission operations team.

The EO-1 domain models the operations of the EO-1 operations for a two-day horizon [13]. It consists of 14 resources, 10 state variables and total of 38 different activity types. Each EO-1 problem instance includes a randomly generated, fixed profile that represents typical weather and instrument pattern. Each problem also includes 3 to 16 randomly placed instrument requests for observations and calibrations, and between 50 and 175 communications satellite passes.

The preferences for EO-1 include preferences for more calibrations and observations, earlier start times for the observations, fewer solar array and aperture manipulations, lower maximum value over the entire horizon for the solar array, and higher levels of propellant.

Applying the quantile-quantile (Q-Q) test to the EO-1 hypotheses shows that they are very likely normal distributions. The Q-Q test compares the quantiles of the samples with a normal distribution, and departures in linearity of the resulting plot show how the samples differ from a normal distribution. Results of applying the Q-Q test to these three domains is shown in [1].

Figures 4 and 5 show scores of the generated heuristic combinations over 35 cycles of the search algorithms. Although the curves for the scores of the two different search algorithms are different, the percentage of improvement for the high scoring hypothesis within each cycle is similar (128% for the linear search compared with 147% for the genetic algorithm). The percentage improvement for the mean score is somewhat greater, 161% for the genetic algorithm compared with 116% for the linear search. The high scoring heuristic combinations are also somewhat different: the local search hypotheses use a significantly lower percentage of random heuristics than the genetic algorithm hypotheses, illustrating two different local maxima in the search space.

Identical runs in all of these domains using the expected loss criterion for the adaptive problem solver yielded very similar results using a similar number of iterations per cycle of adaptive problem solving, so results using expected loss requirements have been omitted.

New Millennium Space Technologies Four Landed Operations Domain– The ST-4 domain models the landed operations of a spacecraft designed to land on a comet and return a sample to earth. This model has 6 shared resources, 6 state variables, and 22 activity types. Resources and states include battery level, bus power, communications, orbiter-in-view, drill location, drill state, oven states for a primary and backup oven state, camera state, and RAM. There are two activity groups that correspond to different types of experiments: mining and analyzing a sample, and taking a picture. Each ST-4 problem instance includes a randomly generated, fixed profile that represents communications visibility to the orbiting spacecraft. Each problem also includes between 1 and 11 mining activities and between 1 and 24 picture experiments at random start times.

The preferences for ST-4 include more imaging, more mining, more battery power over the planning horizon, fewer drill movements, and fewer uplinks.

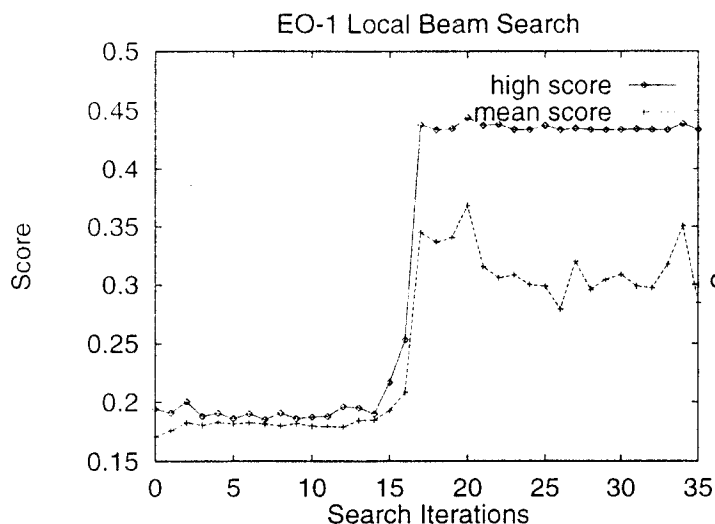


Figure 4: EO-1 model search iteration maximum and average scores for 35 iterations of the local beam search (beam = 2), using rational PAC as the requirement for the adaptive problem solver.

Based on the Q-Q test, hypotheses from the ST-4 domain are likely to be normally distributed, and thus provides a good model for adaptive problem solving [1]. Graph 6 shows the mean and high scores of the generated heuristic combinations over 25 cycles of the search algorithms. Although the indifference ratio for the PAC algorithm is three times higher than in EO-1, the score rises significantly from the starting vector. The high score reaches a maximum improvement of 14%, and the mean score has a maximum improvement of 18%.

This domain includes an expert-generated heuristic for choosing a method to resolve schedule conflicts in a manner appropriate to the DS-4 model, such as moving activities as opposed to deleting them when a resource has been overcommitted. It is interesting to note that even in the best set of hypotheses, the average usage of this heuristic was only 22% for a choice point with only 3 possible heuristics, and no hypotheses with the expert heuristic usage over 40% was ranked in the top third in the adaptive problem solver's ranking.

Rocky-7 Mars Rover Domain – Rocky-7 is a prototype Mars rover for long-range planetary science gathering. The rover domain models operations of a prototype rover for a typical Martian day [10]. It consists of 18 shared resources, 12 state variables and 32 activity types. Resources and states include cameras (front, rear, mast), mast, shovel, spectrometer, solar array, battery, and RAM. There are three activity types that correspond to different types of science experiments: digging at a location, collecting a spectrometer reading from target, and taking an image from a location (panorama, front, rear). Rover problems are constructed by generating between 1 to 12 experiments and randomly generating parameters for the experiments (such as target locations). Heuristics include traveling salesman heuristics which attempt to order the rover moves such that the total distance traveled is minimized.

Rocky-7 preferences include preferences for more science activities and earlier start times for those activities, less traversing and earlier start times for traversals, less battery usage,

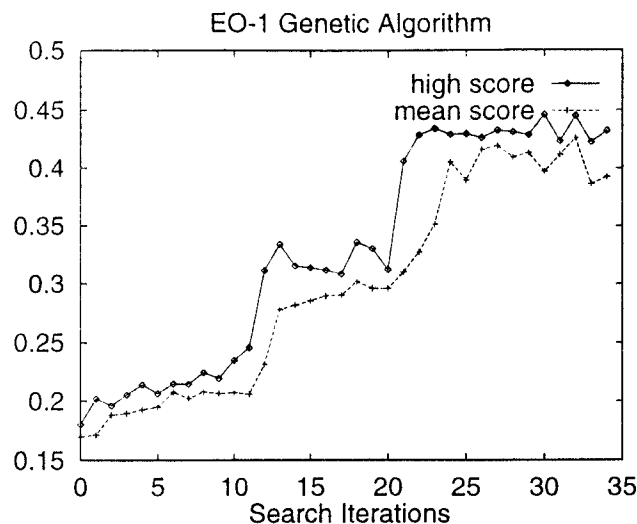


Figure 5: EO-1 model search iteration maximum and average scores for 35 iterations of the genetic algorithm search, using rational PAC as the requirement for the adaptive problem solver.

fewer mast manipulations, and less time that the mast is deployed over the planning horizon.

Graphs 7 show scores of the generated heuristic combinations over 40 cycles of the search algorithms. The graph describing the genetic algorithm search for the Rocky-7 domain was omitted because of its similarity to the linear search graph.

The hypotheses in the Rocky-7 domain appear to come from a non-normally distributed distribution compared to both EO-1 and ST-4, as shown by applying the Q-Q test to the original hypothesis [1]. The adaptive problem solver decision requirement assumes a normal distribution, and the Rocky-7 results illustrates the problem with violating this assumption. Violating the assumption of normality leads to evaluations which cannot provide strong statistical guarantees as to their accuracy. The hypotheses for this particular Rocky-7 problem appear to be less continuous than the domain for EO-1. Over all the search iterations, the greatest improvement in the max scores 101%, and the greatest improvement in the mean score is 101%, although the accuracy of the evaluations is not guaranteed because of the violated normality assumption.

The five heuristics in the set that were designed for the Rover domain by experts, including the multiple traveling salesman path planning heuristics, peak at 45%, 38%, 35%, 34%, and 9% usage for each of their specific choice points, over all of the top hypotheses chosen by the adaptive problem solver. This might indicate that these heuristics should be used in moderation with this domain instance.

RELATED WORK

Evaluating control strategies is a growing research topic. Horvitz originally described a method for evaluating algorithms based on a cost versus quality tradeoff [7]. Russell, Subramanian, and Parr used dynamic programming to rationally select among a set of control strategies by estimating utility (which includes cost) [12]. The MULTI-TAC system considers all k -wise combinations of heuristics for solving a CSP in its evaluation which also avoids problems with local

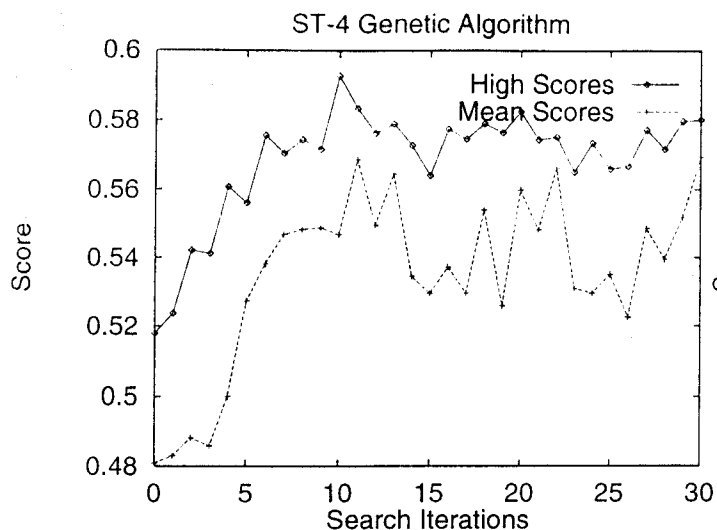


Figure 6: ST-4 model maximum and average scores by iteration for 30 genetic algorithm generations, using rational PAC as the requirement for the adaptive problem solver.

maxima, but at a large expense to the search [8]. Fink describes a method that sets time bounds for selection as opposed to parameter estimation accuracy, since sampling time is not large enough to attempt to minimize the number of samples [3]. Previous articles describing adaptive problem solving have developed general methods have been developed for transforming a standard problem solver into an adaptive one [Gratch & DeJong 1992, 6], illustrated the application of adaptive problem solving to real world scheduling problems [6], and showed how adaptive problem solving can be cast as a resource allocation problem [1]. We expand on these topics by evaluating different methods for generating hypotheses which can be used in adaptive problem solving to efficiently estimate their utility and cost, considered separately.

FUTURE WORK

In the area of adaptive problem solving, additional work has been proposed for the stopping criteria, which can be resource bounded (specifically, time as a resource) instead of a relaxation of the ranking requirement, as in previous works on similar topics [3]. Different methods of combining heuristics could be applied to problems of this type. One method is composite strategies, from operations research, which involve logical decisions about the relative usage of heuristics as opposed to statistical methods. Another method is a portfolio approach, which combines heuristics in a method similar to a financial portfolio.

Current results do not indicate any direct benefit to using either local beam search or genetic algorithms over the alternative. In order to predict an effective search algorithm for each environment, it would be useful to generate a landscape of the utilities for the hypothesis space [16]. Previous work has been done in deterministic landscape generation [16, 15], but no practical work has been done in stochastic landscape generation, which is what this domain requires.

More intelligent methods of searching over the space of hypotheses could be exploited in this domain. It is not clear that all of the model domains are continuous, so a further study of

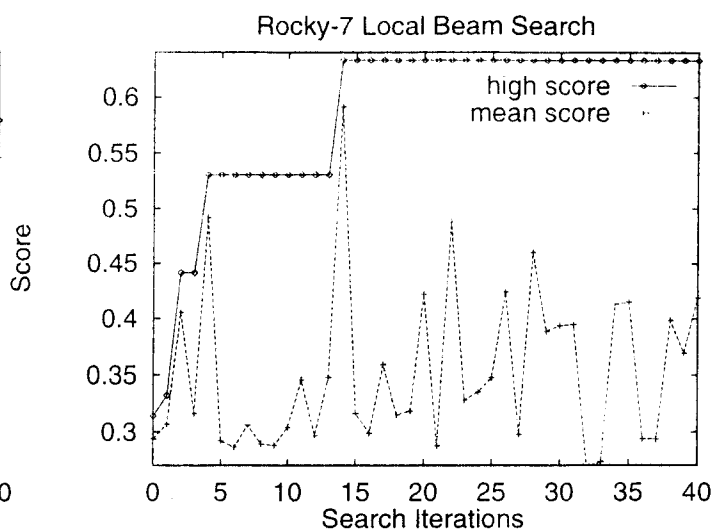


Figure 7: Rocky-7 model highest maximum and search iteration average scores for 40 iterations of the local beam search, using rational PAC as the requirement for the adaptive problem solver.

the shape of the domain should precede the choice of a search method. At a lower level, changing the mutation operators in the current algorithms, such as intentionally weighting one heuristic heavily out of all of the heuristics for a choice point, may direct the search more efficiently.

CONCLUSIONS

This paper outlines different methods for generating control strategies to use in adaptive problem solving, with the goal of finding a control strategy or set of control strategies that performs well in the given planning and scheduling environment. The idea of rational allocation is discussed along with the statistical methods behind an adaptive problem solver. The purpose is validated in all three planning and scheduling domains, by showing significant overall improvement in the generated plans. Two hypothesis generation techniques were explored based on the the amount and types of improvement they allowed.

Empirically, it appears that these methods could be used in a mission operations environment to generate and evaluate a domain-specific set of heuristics to control automated planning and scheduling, either on- or off-board the spacecraft. These results are significant in that autonomous spacecraft planning and scheduling is becoming a realistic option for missions to unknown environments.

ACKNOWLEDGMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] Chien, S.; Gratch, J.; Engelhardt, B.; Mutz, D.; Stechert, A. 1999. Improving Statistical Hypothesis Selection Using Rational Resource Allocation. In *Journal of Artificial Intelligence Research* (to appear).
- [2] Chien, S.; Gratch, J.; Burl, M. 1995. On the Efficient Allocation of Resources for Hypothesis Evaluation: A Statistical Approach.

In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*.

- [3] Fink, E. 1998. How to Solve it Automatically: Selection among Problem-Solving Methods. In *Proceedings of the Fifth International Conference of AI Planning Systems*, 128-136.
- [4] Fukunaga, A.; Rabideau, G.; Chien, S.; and Yan, D. 1997. Towards an application framework for automated planning and scheduling. In *Proceedings of the 1997 International Symposium on Artificial Intelligence, Robotics and Automation for Space*.
- [5] Goldberg, D. 1989. *Genetic Algorithms: In Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley.
- [6] Gratch, J.; DeJong, G. 1996. A Decision-theoretic Approach to Solving the EBL Utility Problem. In *Artificial Intelligence*.
- [Gratch & DeJong 1992] Gratch, J.; DeJong, G. 1992. COMPOSER: A Probabilistic Solution to the Utility Problem in Speed-up Learning. In *Proceedings of the Tenth National Conference on Artificial Intelligence*.
- [7] Horvitz, E. 1988. Reasoning under Varying and Uncertain Resource Constraints. *Proceedings of the Seventh National Conference on Artificial Intelligence*, 111-116.
- [8] Minton, S. 1996. Automatically Configuring Constraint Satisfaction Programs: A Case Study. In *Constraints* 1:1(7-43).
- [9] Minton, S. 1988. Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems. *Artificial Intelligence* 58:161-205.
- [10] Rabideau, G.; Chien, S.; Backes, G.; Chalfant, G.; Tso, K. 1999. A Step Towards an Autonomous Planetary Rover. *Space Technology and Applications International Forum*.
- [11] Russell, S.; Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.
- [12] Russell, S.; Subramanian, D.; and Parr, R. 1993. Provably Bounded Optimal Agents. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*.
- [13] Sherwood, R.; Govindjee, A.; Yan, D.; Rabideau, G.; Chien, S.; Fukunaga, A. 1998. Using ASPEN to automate EO-1 Activity Planning. In *Proceedings of the 1998 IEEE Aerospace Conference*.
- [14] Smith, S. 1994. OPIS: A Methodology and Architecture for Reactive Scheduling. In Zweben, M. and Fox, M., eds., *Intelligent Scheduling*. San Francisco: Morgan Kaufmann. 29-66.
- [15] Whitley, D.; Mathias, K.; Rana, S.I Dsubera, J. 1995. Building Better Test Functions In *Proceedings of the International Conference on Genetic Algorithms*.
- [16] Wolpert, D.H.; Macready W.G. 1996. No Free Lunch Theorems for Search *SFI-TR-95-02-010*.
- [17] JPL 1999. <http://www.jpl.nasa.gov/missions/>.
- [18] Zweben, M.; Daun, B.; Davis, E.; and Deale, M. 1994. Scheduling and rescheduling with iterative repair. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. San Francisco, CA: Morgan Kaufmann. 241-256.

BIOGRAPHY

Barbara Engelhardt is a member of the Artificial Intelligence Group at the Jet Propulsion Laboratory, California Institute of Technology where she performs research and development of automated planning and scheduling systems and statistical hypothesis evaluation. She received a B.S. in symbolic systems in 1998 and an M.S. in computer science in 1999 from Stanford University.

Dr. Steve Chien is Technical Group Supervisor of the Artificial Intelligence Group and Principal Computer Scientist in the Exploration Systems Autonomy Section at the Jet Propulsion Laboratory, California Institute of Technology where he leads efforts in automated planning and scheduling for space exploration. Dr. Chien is also an Adjunct Assistant Professor with the Department of Computer Science of the University of Southern California. He holds B.S., M.S., and Ph.D. degrees in Computer Science, all from the University of Illinois.

Darren Mutz is a member of the Artificial Intelligence Group at the Jet Propulsion Laboratory, California Institute of Technology, where his current projects include Long Range Science Rover, statistical hypothesis evaluation, and automated planning and scheduling. He received the Bachelor of Science degree in Computer Science at the University of California, Santa Barbara in 1997.